Josef Ressel Center for User-friendly Secure Mobile Environments (u'smile)
University of Applied Sciences Upper Austria
FH OÖ Forschungs & Entwicklungs GmbH

Ansprechpartner:
Dr. Michael Roland

Softwarepark 11
4232 Hagenberg/Austria

+43 (0)50804-27149
michael.roland@fh-hagenberg.at
www.usmile.at • www.fh-ooe.at

Software-Firewall zum Schutz vor (Bad)USB-Geräten

Ausgangspunkt

Vor kurzem wurde unter dem Begriff „BadUSB" [1, 2] ein Gefahrenszenario bekannt, bei dem bereits durch das Anschließen eines USB-Geräts ein System angegriffen werden kann. Plug-and-Play-USB-Geräte können beispielsweise Tastatur- und Mauseingaben emulieren, um so automatisiert Befehle auszuführen sobald ein solches Gerät angeschlossen wird. Gleichzeitig könnte ein solches Gerät auch ein Laufwerk (Massenspeicher) oder eine Netzwerkkarte emulieren, um Dateien ins System einzuschleusen. Ein Angreifer kann das Ausnutzen um automatisiert Schadsoftware zu starten oder Netzwerkeinstellungen so zu manipulieren, das Netzwerkverbindungen abgehört werden können.

BadUSB ist kein neues Konzept [3]. Bereits vor BadUSB bekannt wurde, gab es Werkzeuge wie USB Rubber Duck [4], mit denen Tastatureingaben, … in ein System eingeschleust werden können. Das besondere an BadUSB ist das erhöhte Gefahrenpotential: Die Firmware von gewöhnlichen USB-Geräten (z.B. USB-Massenspeichern) kann (direkt über die USB-Schnittstelle selbst) für Angriffe modifiziert werden.

Diese Klasse von Angriffsmöglichkeiten stellt ein grundsätzliches Problem bei Mehrzweck-Plug-and-Play-Schnittstellen wie USB dar. Der Benutzer kann beim Betrachten des physischen Geräts nur Vermutungen über dessen tatsächlichen Funktionalität anstellen. Zum Beispiel wurde ein Benutzer von einer Webcam erwarten, sich diese als Webcam (und nur als solche) am System anmeldet. Die einzige zuverlässige Methode, um herauszufinden, welche Funktionalität ein Gerät tatsächlich bereitstellt, ist, dieses an ein System anzuschließen und die tatsächlich bereitgestellten Gerätefunktionen zu analysieren. Allerdings wird das Gerät genau dabei, also beim Anschließen, automatisch aktiviert und im System bereitgestellt. Folglich hat der Benutzer keine Gelegenheit ein Gerät nach dem Anschließen zu analysieren ohne, dass gleichzeitig eventuell vorhandene bösartige Funktionen aktiviert werden.

Aufgabenstellung

Um einen Schutz gegen solche Angriffsszenarien zu bieten soll eine Software-Firewall entwickelt werden (vgl. [2, 6]. Standardmäßig sollen alle (neu erkannten) USB-Geräte blockiert werden. Bei der Detektion neuer Geräte soll die weitere Vorgehensweise anhand einer Benutzerabfrage entschieden werden. Dem Benutzer soll dabei klar dargestellt werden welche Funktionen ein Gerät bereitstellt. Erst nachdem ein Gerät vom Benutzer freigegeben wurde, soll dieses im System aktiviert werden.

Interessante Fragestellungen:

- Wie lässt sich diese Art von Firewall/Unterbrechung der Plug-and-Play-Funktion auf Windows und Linux implementieren?
- Welche Informationen sollten dem Benutzer angezeigt werden, um die Entscheidung über die Akzeptanz eines Gerätes zu erleichtern? Z.B.
    - „Dieses Gerät ist ein USB-Massenspeicher."
    - „Dieses Gerät war zuvor an einem anderen USB-Port angeschlossen."
    - „Dieses Gerät wurde bereits in der Vergangenheit benutzt."
    - „Dieses Gerät ist eine Tastatur, es ist jedoch bereits eine andere Tastatur am System angeschlossen."
- Wie können diese Informationen sinnvoll und benutzerfreundlich dargestellt werden, damit Warnungen der Firewall nicht einfach ignoriert werden?
- Wie kann ein Whitelisting von bereits erkannten und freigegebenen Geräten implementiert werden? Z.B.
    - Sind Geräte eindeutig identifizierbar (oder könnten solche Informationen gezielt gefälscht werden)?
    - Können Geräte bestimmten USB-Anschlüssen eindeutig zugeordnet werden? Über zwischengeschaltete USB-Hubs hinweg?

Themenfelder

- USB
- Betriebssysteme und Treiber
- Benutzerinteraktion

Literatur

[1] K. Nohl, S. Krißler, and J. Lell: "BadUSB – On accessories that turn evil", Presented at Black Hat USA 2014, https://srlabs.de/blog/wp-content/uploads/2014/07/SRLabs-BadUSB-BlackHat-v1.pdf

[2] K. Nohl, S. Krißler, and J. Lell: "BadUSB – On accessories that turn evil", Presented at PacSec 2014, https://srlabs.de/blog/wp-content/uploads/2014/11/SRLabs-BadUSB-Pacsec-v2.pdf

[3] J. Edge: "BadUSB: Clever but not novel", in LWN.net, Aug. 2014, http://lwn.net/Articles/608503/

[4] USB Rubber Duck, https://hakshop.myshopify.com/collections/usb-rubber-ducky

[5] USBdriveby, http://samy.pl/usbdriveby/

[6] Comments on article "Plug-and-play sanitization of USB thumb drives", in LWN.net, Dec. 2014, http://lwn.net/Articles/626559/#CommAnchor626763